IN THE CLAIMS:

EBK 2/10/2004

1-~~22.~~ 21 (Cancelled)

22. (Original) A framework for loading class data structures prior to execution and for resolving called Java® methods, said framework preferentially resolving said called Java® methods as cloned versions of Java® methods within a compilation unit common to a calling Java® method, said framework resolving respective called Java® methods outside said common compilation unit in the event of a version conflict between said respective cloned and external Java® methods.

23. (Original) The framework of claim 22, wherein said version conflict is determined with respect to at least one of a timestamp, a cyclic redundancy check (CRC) and a version control identifier.

24. (Original) The framework of claim 22, wherein said internal constant resolution entries are compiled to produce in-line executable code.

25. (Original) The framework of claim 22, an executing Java® method is provided addressability to a runtime version of its entry in a container class method table.

26. (Original) The framework of claim 23, wherein if a constant pool entry provided by said calling Java® method is to be resolved to a clone class, said framework performs the steps of:

loading said clone class; and

modifying said loaded clone class to represent the respective clone and parent classes for said constant pool entry.

27. (Original) The framework of claim 26, wherein said step of modifying comprises the steps of overlaying a plurality of fields within said clone class to represent corresponding structures of said parent class.

28. (Original) The framework of claim 26, wherein a determination of whether said constant pool entry provided by said calling Java® method is to be resolved to a clone class is made by performing the steps of:

extracting a corresponding constant pool entry pointer;

resolving the constant pool entry to its class; and

determining if the constant pool entry has been resolved to a clone class.